Lab 3

Reg. No: 19BCE1209

Name: Gautam Sanjay Wadhwani

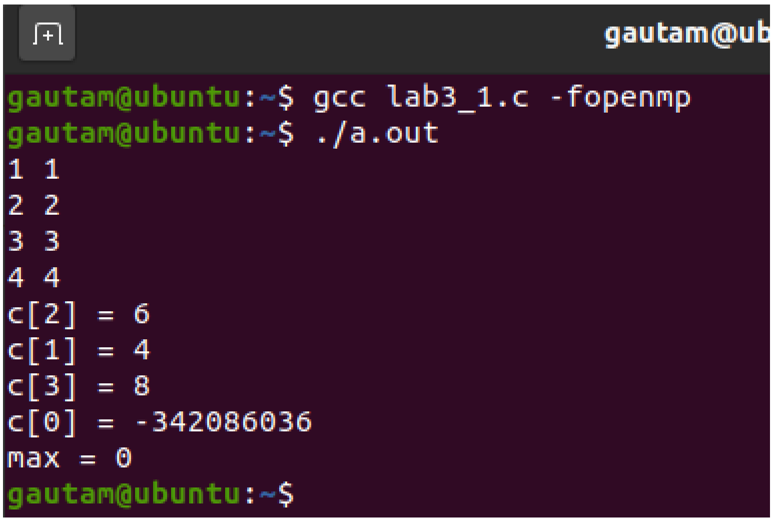
Course: CSE4001 Parallel and Distributed Computing

Q1. Sample first private

Code:

```
#include <stdio.h>
#include <omp.h>

void main() {
   int n = 4, d = 5, max;
   int a[n], b[n], c[n];
   for(int i = 0; i < n; i++) {
          scanf("%d %d", &a[i], &b[i]);
   }
   omp_set_num_threads(4);
   #pragma omp parallel for private(d) firstprivate(max)
   for(int i = 0; i < n; i++) {
          c[i] = a[i] + b[i] + d;
          max = c[i];
          printf("c[%d] = %d\n", i, c[i]);
   }
   printf("max = %d\n", max);
}
```

Output:



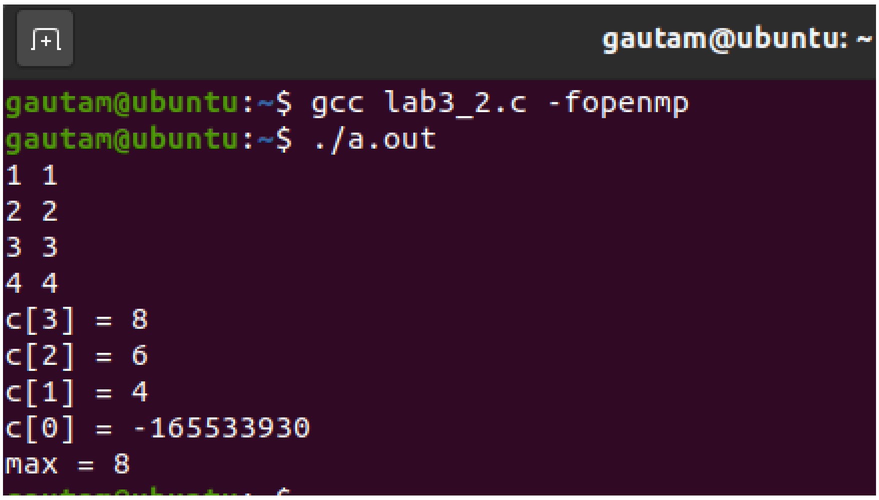
```
gautam@ubuntu:~$ gcc lab3_1.c -fopenmp
gautam@ubuntu:~$ ./a.out
1 1
2 2
3 3
4 4
c[2] = 6
c[1] = 4
c[3] = 8
c[0] = -342086036
max = 0
gautam@ubuntu:~$
```

Q2. Sample last private

Code:

```
#include <stdio.h>

#include <omp.h>


void main() {

   int n = 4, d = 5, max;

   int a[n], b[n], c[n];

   for(int i = 0; i < n; i++) {

          scanf("%d %d", &a[i], &b[i]);

   }

   omp_set_num_threads(4);

   #pragma omp parallel for private(d) lastprivate(max)

   for(int i = 0; i < n; i++) {

          c[i] = a[i] + b[i] + d;

          max = c[i];

          printf("c[%d] = %d\n", i, c[i]);

   }

   printf("max = %d\n", max);
}
```

Output:



```
gautam@ubuntu:~$ gcc lab3_2.c -fopenmp
gautam@ubuntu:~$ ./a.out
1 1
2 2
3 3
4 4
c[3] = 8
c[2] = 6
c[1] = 4
c[0] = -165533930
max = 8
```

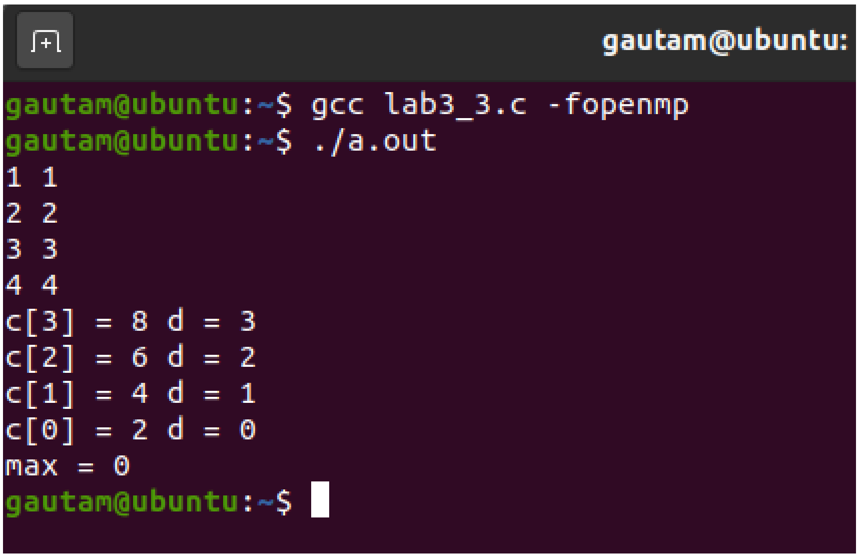
Q3. Sample first private with modification to d variable

Code:

```
#include <stdio.h>

#include <omp.h>


void main() {

   int n = 4, d = 5, max;

   int a[n], b[n], c[n];

   for(int i = 0; i < n; i++) {

            scanf("%d %d", &a[i], &b[i]);

   }

   omp_set_num_threads(4);

   #pragma omp parallel for private(d) firstprivate(max)

   for(int i = 0; i < n; i++) {

            c[i] = a[i] + b[i] + d;

            d += i;

            max = c[i];

            printf("c[%d] = %d d = %d\n", i, c[i], d);
```

```
    }
    printf("max = %d\n", max);
}
```

Output:



gautam@ubuntu:

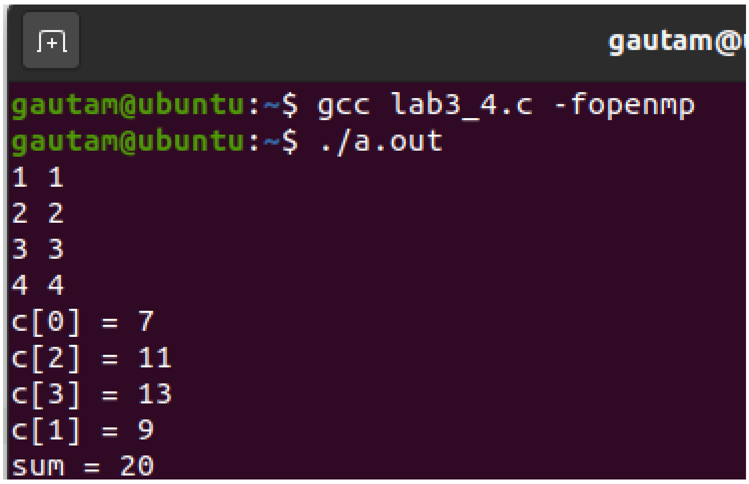
```
gautam@ubuntu:~$ gcc lab3_3.c -fopenmp
gautam@ubuntu:~$ ./a.out
1 1
2 2
3 3
4 4
c[3] = 8 d = 3
c[2] = 6 d = 2
c[1] = 4 d = 1
c[0] = 2 d = 0
max = 0
gautam@ubuntu:~$
```

Q4. question 4 of PDC Lab 3

Code:

```
#include <stdio.h>
#include <omp.h>

void main() {
    int n = 4, d = 5, sum, x = 0;
    int a[n], b[n], c[n];
    for(int i = 0; i < n; i++) {
            scanf("%d %d", &a[i], &b[i]);
    }
    omp_set_num_threads(4);
    #pragma omp parallel for shared(x, d) lastprivate(sum)
    for(int i = 0; i < n; i++) {
            c[i] = a[i] + b[i] + d;
            x += c[i];
            sum = x;

            printf("c[%d] = %d\n", i, c[i]);

    }

    printf("sum = %d\n", sum);

}
```

Output:



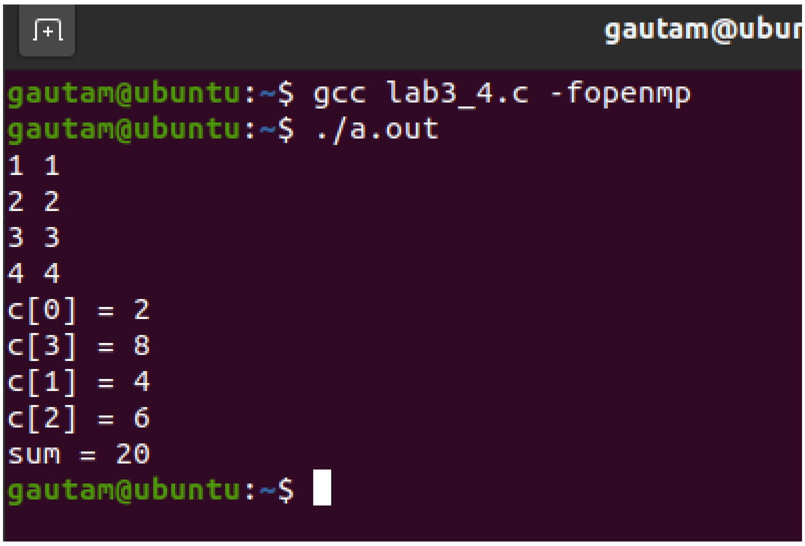
```
gautam@ubuntu:~$ gcc lab3_4.c -fopenmp
gautam@ubuntu:~$ ./a.out
1 1
2 2
3 3
4 4
c[0] = 7
c[2] = 11
c[3] = 13
c[1] = 9
sum = 20
```

Q5. Sum of array of c[i]

Code:

```
#include <stdio.h>

#include <omp.h>


void main() {

    int n = 4, sum, x = 0;

    int a[n], b[n], c[n];

    for(int i = 0; i < n; i++) {

            scanf("%d %d", &a[i], &b[i]);

    }

    omp_set_num_threads(4);

    #pragma omp parallel for shared(x) lastprivate(sum)

    for(int i = 0; i < n; i++) {

            c[i] = a[i] + b[i];

            x += c[i];
            sum = x;

            printf("c[%d] = %d\n", i, c[i]);

    }

    printf("sum = %d\n", sum);

}
```

Output:



```
gautam@ubuntu:~$ gcc lab3_4.c -fopenmp
gautam@ubuntu:~$ ./a.out
1 1
2 2
3 3
4 4
c[0] = 2
c[3] = 8
c[1] = 4
c[2] = 6
sum = 20
gautam@ubuntu:~$
```